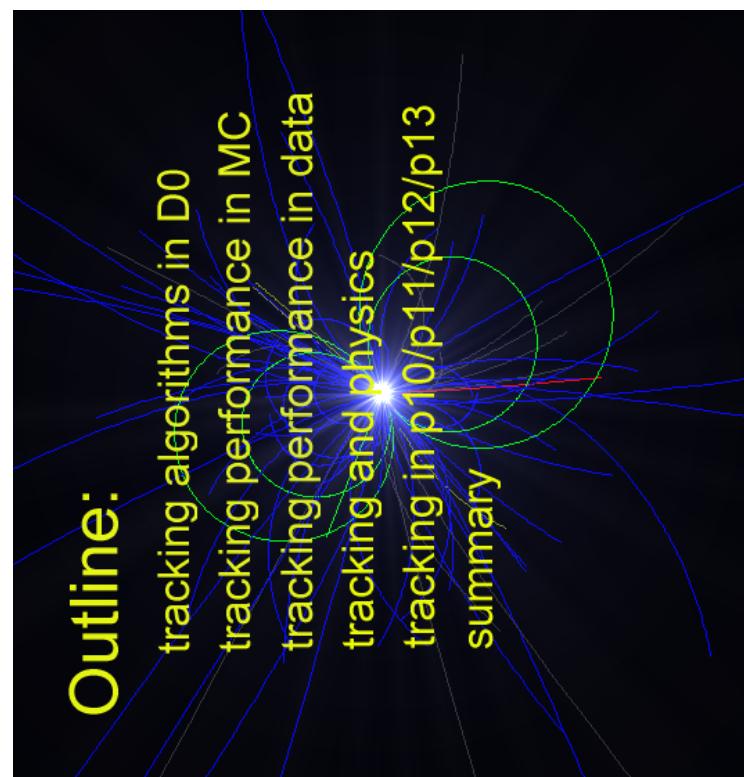




# Tracking in D $\emptyset$

## Global Tracking Group



Andrew Haas  
Avto Kharchilava  
Christian Schmitt  
Dave Adams  
Gavin Hesketh  
Gennadii Borisov  
Herb Greenlee  
Kirstein Goldmann  
Marian Zdrrazil  
Mike Hildreth  
Sasha Khanov  
Slava Kulik  
Valentin Kuznetsov

Prepared by Valentin Kuznetsov  
Univ. of California, Riverside

D $\emptyset$  Workshop, Oklahoma, July 8-12, 2002



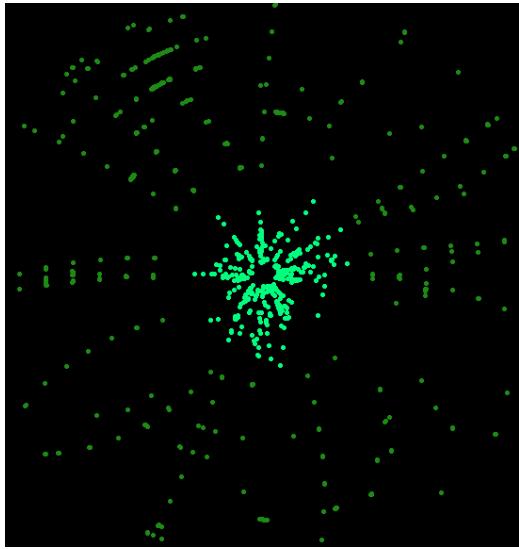
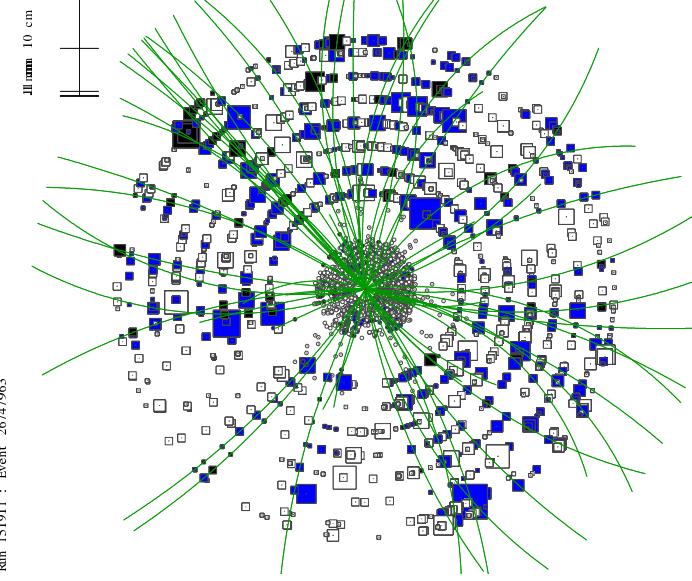
## Introduction

Problem: find association between hits (detector measurements) and construct out of them tracks (particle trajectories)

Solution: human scanning, track algorithms

Requirements: fast, efficient algorithm

Complications: low momentum tracks, scattering, noise, high track density in jets, etc.



You need to balance between CPU, reconstruction time and efficiency vs. scattering, noise,  $p_T$  threshold

Consequences: many techniques have been developed and adapted in HEP community

Experiments try to find the best algorithm for their detector



## DØ implementation of Global TRacking software (GTR) —

Object oriented approach (C++ language) and modular design of DØ software allows development of many tracking algorithms

May be run simultaneously/separately to achieve best performance for different physics tasks

Currently, four algorithms are under investigation:

- Road approach (default algorithm): based on TRF++ uses specific paths (roads) during track finding

Author(s): GTR group

- HTF (histogramming track finder): divides DØ detector into slices in  $(\varphi, \rho)$  and uses Hough transform to reduce initial number of combinations. Existing algorithm can use either CFT, SMT hits or combinations of them to construct tracks

Author: S. Khanov

- Elastic reco (elastic-template algorithm): can use existing tracks (from other track finders) as initial seeds, existing vertex (run after of GTR and vertex code) and/or run in stand-alone mode (construct own seeds and vertices)

Author: A. Haas

- AA (alternative algorithm): starts from 3 hits in different superlayers (SMT or CFT). Track candidates are extended towards CFT.

Author: G. Borissov

Having a variety of track reconstruction algorithms allows us to cross-check the tracking performance in DØ detector on MC and real data



## — Tracking Algorithms (alternative algorithm and road approach) —

**Road Approach:** tracks are searched within a road

Track parameters updated using Kalman filter

Hit association with the track is based on  $\chi^2$   
Used in both fixed target and collider experiments

**Advantages:** very flexible, proven, quite efficient, allow CFT/SMT misses  
**Disadvantages:** requires many roads to be efficient, number of operations grows quickly with number of hits

**Alternative Algorithm:** try all hit combinations to construct track candidates, starts from 3 hits either in different SMT or CFT superlayers

Can find either SMT, CFT or SMT/CFT tracks

Used for alignment procedure

**Advantages:** no  $p_T$  restriction (e.g. very attractive for low- $p_T$  B physics), not restricted to interaction point

**Disadvantages:** not yet implemented in framework, need to be tested on various physics samples



## — Tracking Algorithms (histogramming method) —

**Histogramming Method:** based on Hough transform which translates a single hit into a line in  $(\varphi, \rho)$  space

Lines from all hits on a track intersect at one point corresponding to the vector of track parameters

Those lines make bands in a histogram whose peaks are track templates

Each template is a track candidate consisting of several hits, with approximately known track parameters. The templates are further processed applying the Kalman filtering. During this stage, fake templates are discarded, wrong hits removed, and track parameters accurately calculated

Used mostly in collider experiments

**Advantages:** number of operations proportional to number of hits, small number of candidates within a histogram bin

**Disadvantages:** sensitive to vertex position, assumes small impact parameter and therefore is not sensitive to  $K_s$



## Tracking Algorithms (elastic arms)

Elastic-Template Approach: based on estimation of track parameters concurrently with assignment of hits

Defines a track-hit probability matrix that is a probability for any hit to be on any track at a given “temperature”

Assignment probability is chosen on the competition basis:

- competition between all hits for each track, but no competition between the tracks
- competition between all tracks for each hit, but no competition between the hits
- global competition between all entries that are incompatible, i.e. belong to the same hit or the same track

Tracking begins with “seeds” filtered from a histogramming technique

Seeds are fit “elastically” they continuously change their probabilities of being connected with each hit as their track-parameters evolve

Hits not originally associated with a track seed can become associated through the elastic fitting process

Developed for LHC/SSC collider experiments, considered for ATLAS

Advantages: equal performance with road-finding algorithm at low track densities, significantly higher performance at high track densities.

Disadvantages: sensitive to vertex position (use own vertex algorithm and/or use existing one), quite significant fake rate (under investigation)



Kalman filter is a common algorithm for prediction and evaluation of track parameters based on known information

A track in space can be described by 5D state vector

Evolution of the state vector is described by a discrete system of linear equations:

$$\mathbf{x}_k = \mathbf{F}_{k-1} \mathbf{x}_{k-1} + \omega_{k-1}$$

which define the change in status of this vector based on the previous measurement point  $\mathbf{x}_{k-1}$

Matrix  $\mathbf{F}_{k-1}$  is the track propagator from measurement  $k-1$  to  $k$  and  $\omega_{k-1}$  describes the random noise of the system

The Kalman filter proceeds by performing three distinct operations:

- ★ *Prediction*, where the status of the state vector is estimated at a future measurement point;
- ★ *Filtering*, where the current estimation of the state vector is carried out based on the previous measurements; and
- ★ *Smoothing*, where the estimation of the state vector at a previous measurement is re-evaluated with the new information from the present measurement.

Path / Algorithms	GTR	HTF	Elastic	AA
SMT stand-alone	230	901	10230	x
CFT axial	113	x	x	x
CFT stereo	111	x	x	x
CFT combined	101	x	x	x
CFT→SMT	201	931	x	-
SMT→CFT	333	911	x	x
Final Fitter	x	x	x	x
Combined	401	911	10401	x

All tracking algorithms are implemented as framework packages

Each algorithm specify an unique chunk ID to allow comparison

All of them use one final track refitting step based on D0Propagator

Global cuts such as  $p_T$  threshold, # misses, etc. in RCP

One common user interface: GTrack

Default algorithm is road following, used for data processing

It is possible to combine different algorithms together, filter tracks, refit them and store to another chunk

Physics groups are encouraged to test/use any tracking algorithm(s) in post-processing analysis ⇒ DØ need to decide which one to use



- GTR packages are released and tested on the farm every week
- Farm usage allows rapid response to various problems with design and implementation of the tracking algorithm(s)

GTR by numbers (release t01.71.00)

	gtr	trf	smt	cft	total
packages	22	23	24	21	90
classes	268	399	252	148	1401
RCP's	120	0	134	80	334
OBS's	5	2	33	39	79

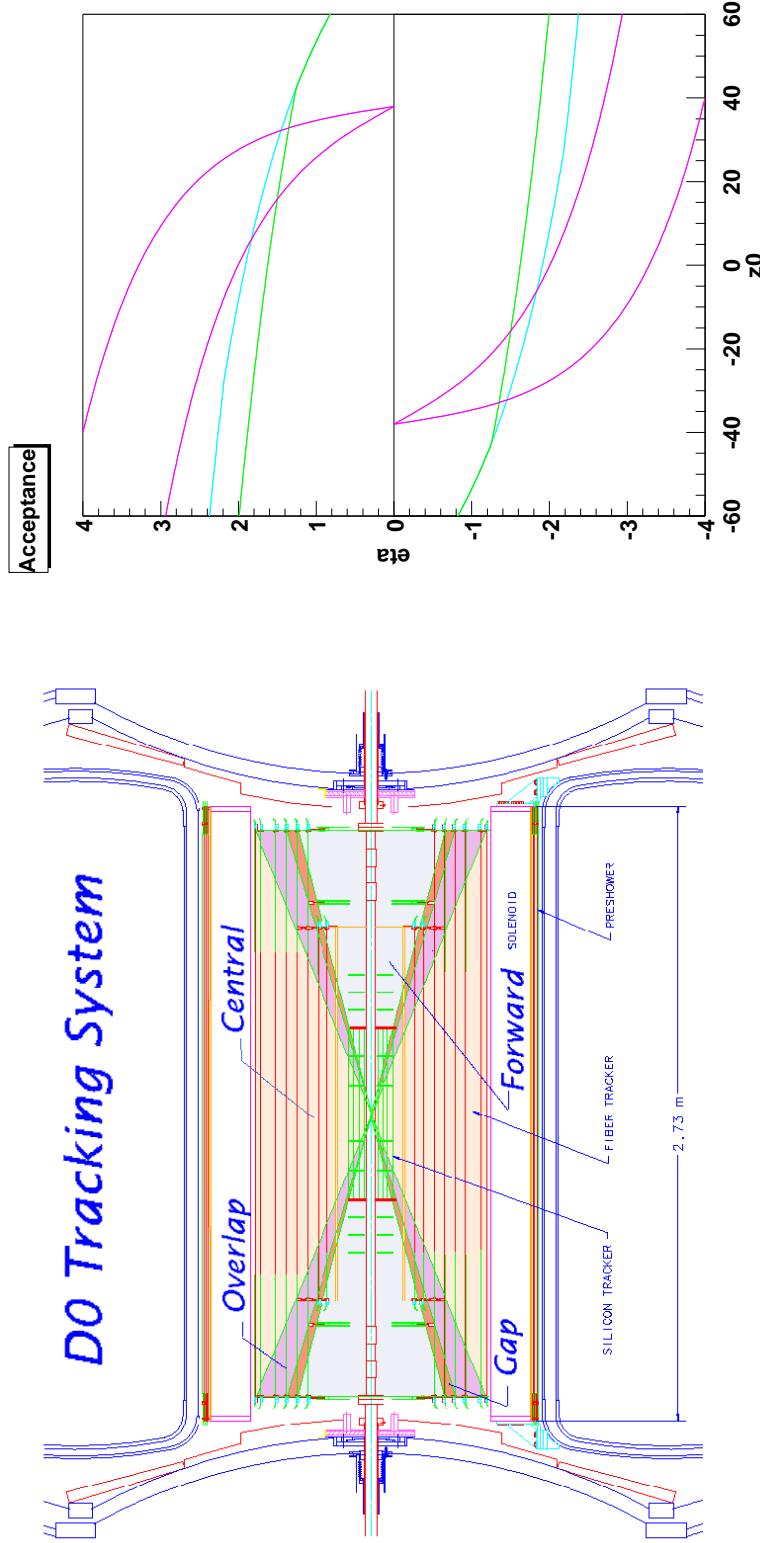
Global cuts (e.g. minimum  $p_T$ , # of misses/track) and objects (e.g. propagators) are constructed once during initialization in ObjTable

Parameters and objects are controlled by RCP (run control parameters) and OBS (object streams) files

GTR extensively uses RCP files for external parameters

GTR uses OBS files to define a class objects on fly. The OBS files contain instructions for track finding, i.e. the list of roads to follow

GTR automatically manage magnetic field sign since p11 release



Divide acceptance into four regions:

- central** - full CFT extended into SMT
- forward** - forward SMT with three F-disks
- overlap** - partial CFT extended into SMT
- gap** - between overlap and forward

A Path for road-following algorithm is constructed for each range by requiring it to cross a particular number of sub detector layers

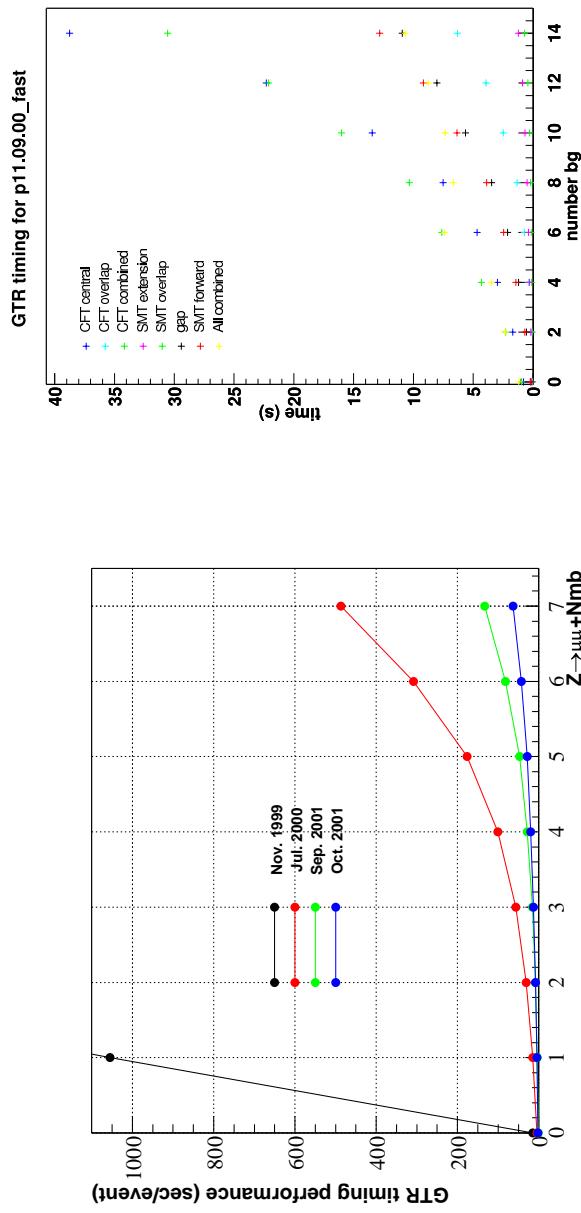


## Tracking Performance

Test global tracking on standard samples of  $Z \rightarrow \mu\mu + Nmb$ ,  $N = 0, \dots, 14$

GTR performance on DØ farm is measured every week

We monitor time usage during each step of track reconstruction program(s)



Total time spent by track reconstruction program for a sample of  $Z \rightarrow \mu\mu + Nmb$

version/Nmb	0mb	2mb	4mb	6mb	8mb	10mb	12mb	14mb
data path (p10.15.01)	1.5	4.2	8.6	14.7	24.5	41.4	63.4	96.3
data path (p11.07.00)	7.6	18.6	36.3	60.1	94.2	158.7	220.5	304.9
data path (p11.09.00)	4.3	9.9	17.4	30.4	40.6	61.6	87.5	127.2
MC path (t02.17.00)	4.3	10.0	18.8	26.8	38.8	56.9	76.9	105.1

Final goal to achieve 10 sec/event

### Definitions:

*Track-finding Performance is assessed by comparing reconstructed and Monte Carlo tracks using kinematic matching*

*Match  $\chi^2$ : average value  $\simeq 5$*

*Nearness: reconstructed tracks are matched to the nearest MC track using the match  $\chi^2$ . The maximum allowed match  $\chi^2 = 500$*

*If multiple reco tracks are matched to one MC track, then all but the closest are left unmatched.*

*Matched MC tracks are *found tracks**

*Found tracks with match  $\chi^2 > 25$  are called misreconstructed. Those below are *well reconstructed**

*Unmatched reco tracks are called *fakes**

### Performance Metrics:

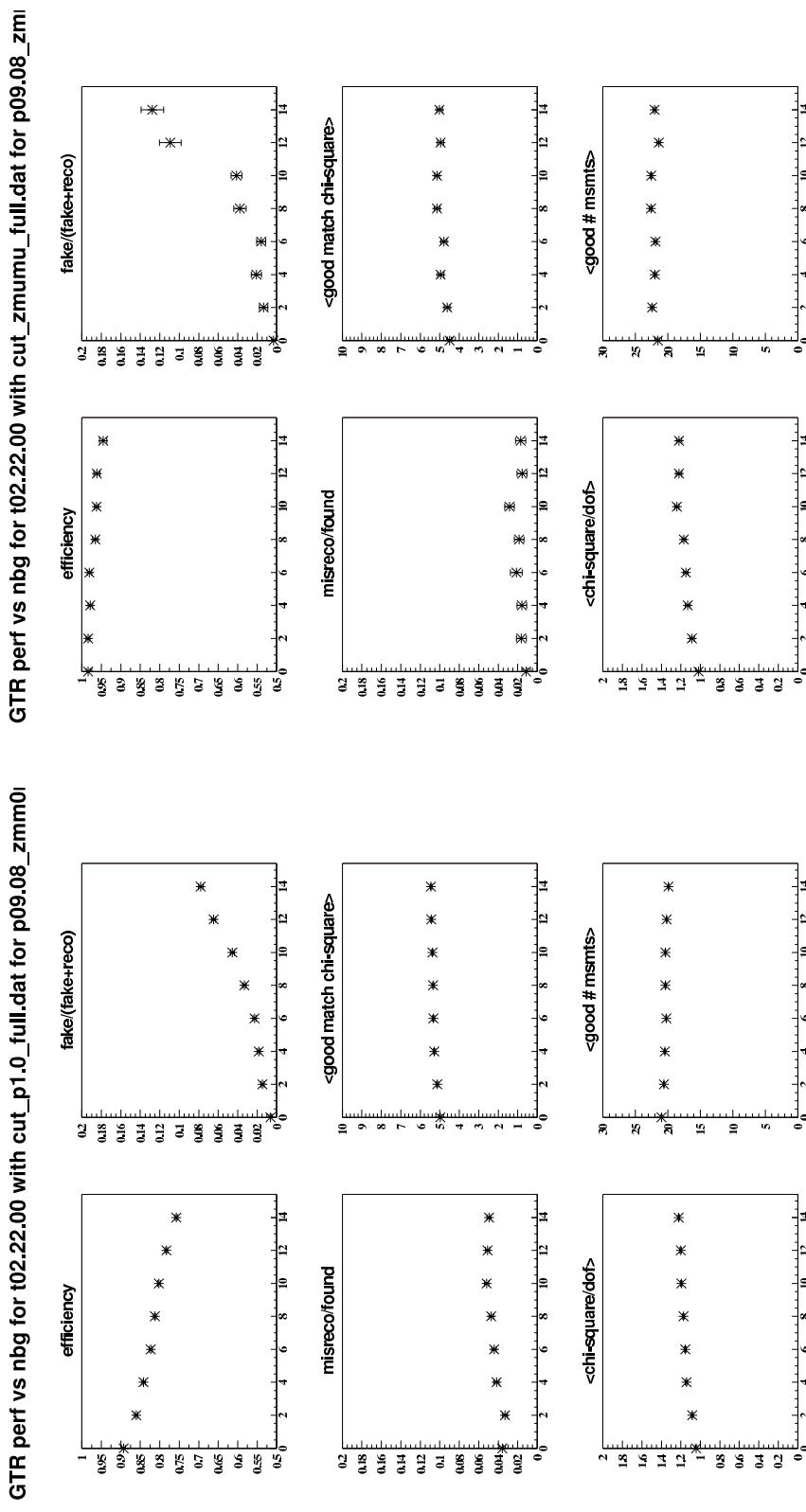
*Efficiency is the fraction of Monte Carlo tracks which are matched*

*Misreco fraction is the fraction of found tracks that are misreconstructed*

*Fake ratio is the number of fake tracks divided by the number of (found+fake) tracks*

**But a hit-by-hit comparison of reco'd and MC tracks is ready**

## Performance of default (road-following) algorithm:

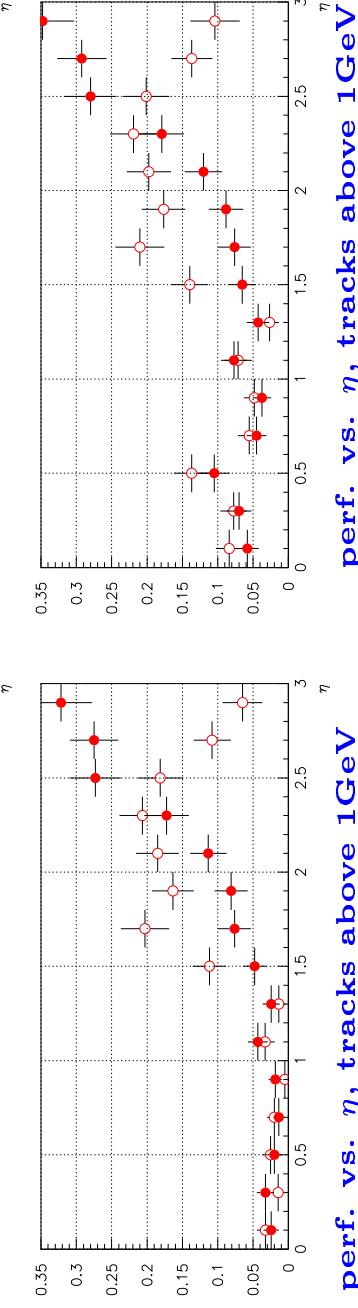
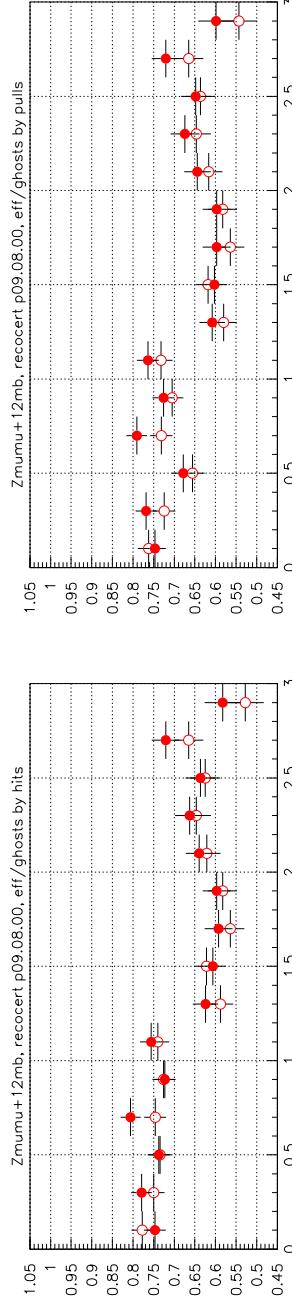


perf. vs. # min. bias events, perf. vs. # min. bias events,  
all tracks above 1GeV  $\mu$ 's from Z above 10GeV

For low min. bias events tracking efficiency above  $\sim 85\%$  for  $p_T \geq 1$  GeV  
and  $\sim 98\%$  for high  $p_T \geq 10$  GeV muons

## Tracking Performance (cont'd)

*Performance of track reconstruction program ( $gtr$  vs.  $htf$ ):*

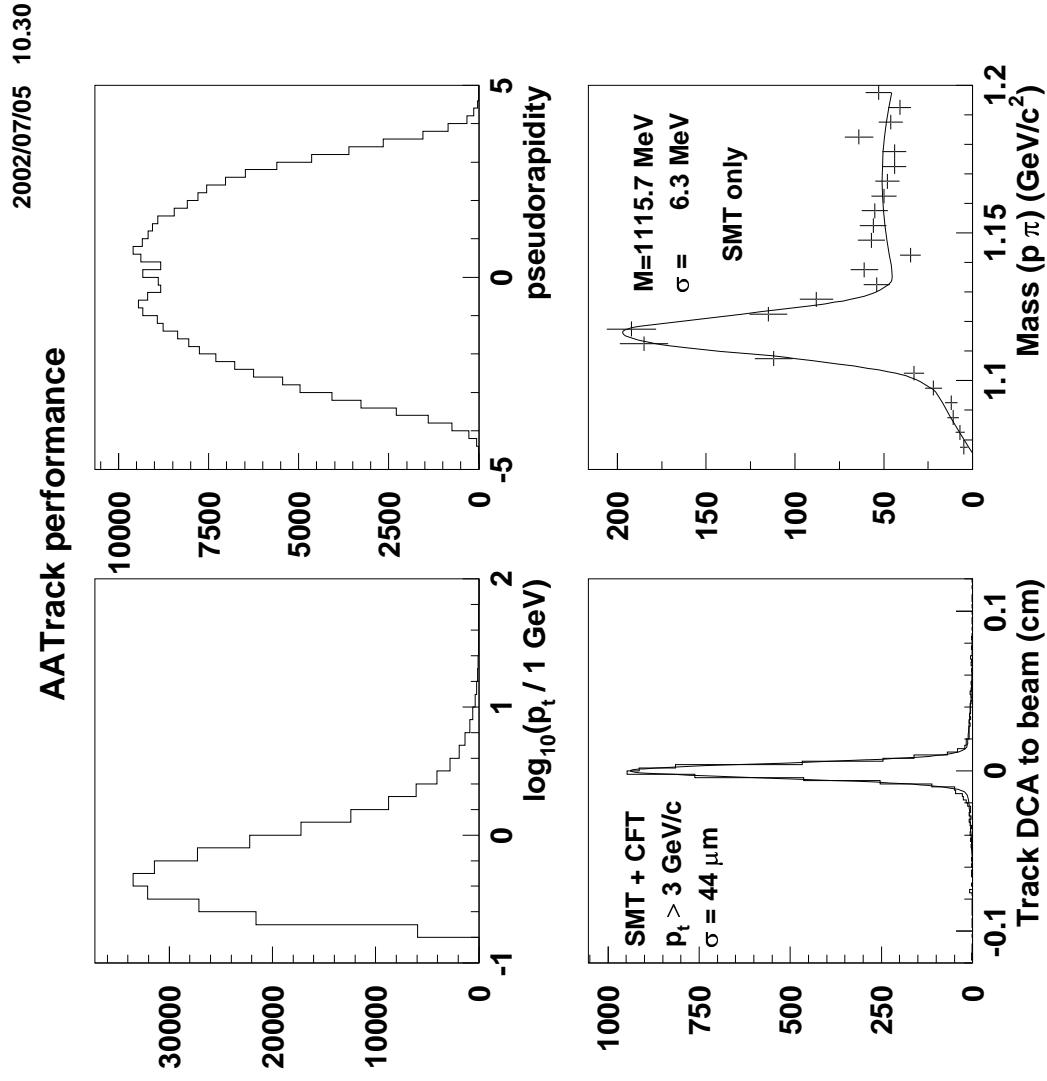


**perf. vs.  $\eta$ , tracks above 1 GeV**  
**hit-by-hit comparison**

*Performance of track reconstruction program ( $gtr$  vs.  $gtr+elastic\_reco$ ):*

- about 30% of the missing tracks (3% overall) are lost due to the sharing or grouping of CFT clusters
- about 50% of the missing tracks (5% overall) are lost because the particle decays before exiting the CFT
- about 20% of the missing tracks (2% overall) are lost due to inefficiencies in the two tracking algorithms
- The addition of elastic\_reco increases the real physics efficiency from 86% to 98% on MC QCD  $p_T > 80$  GeV with 0.5 minbias

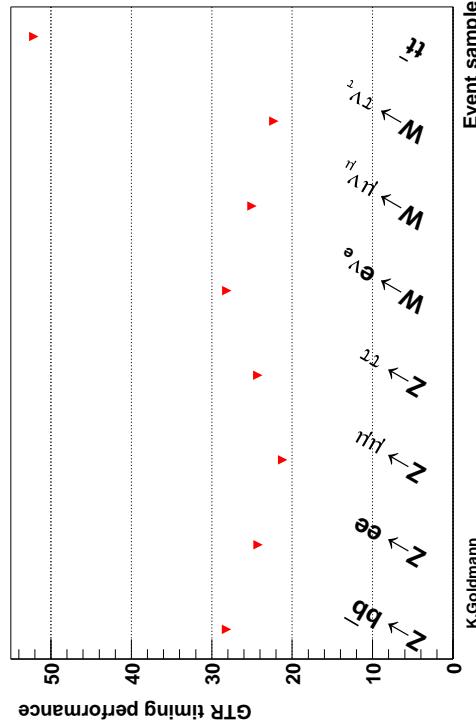
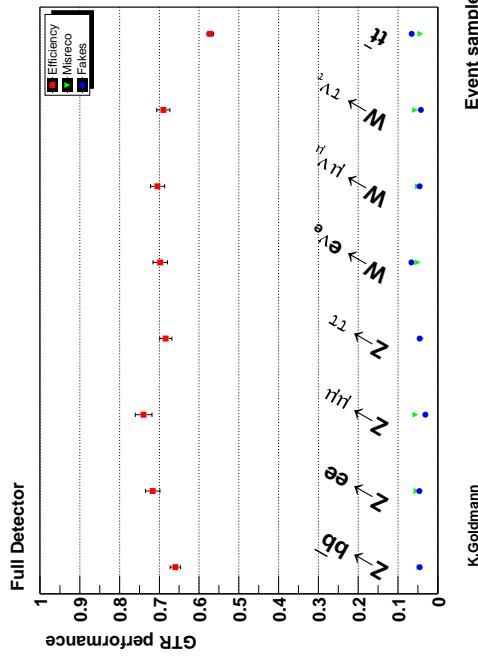
*Performance of alternative tracking (AA):*



A new algorithm has been successfully used for alignment procedure. For more details, see G. Borissov's talk on TTK/alignment/tracking session.

It is expected to appear soon as a framework package (for p13).

- GTR performance is measured for many interesting physics samples ( $Z \rightarrow \ell\ell, b\bar{b}, W \rightarrow \ell\nu_\ell, t\bar{t}$  plus 2.5mb average)
- We achieved quite stable performance, additional tuning may be possible using different algorithms and/or cuts
- Performance is measured for major releases (production and major development branches)



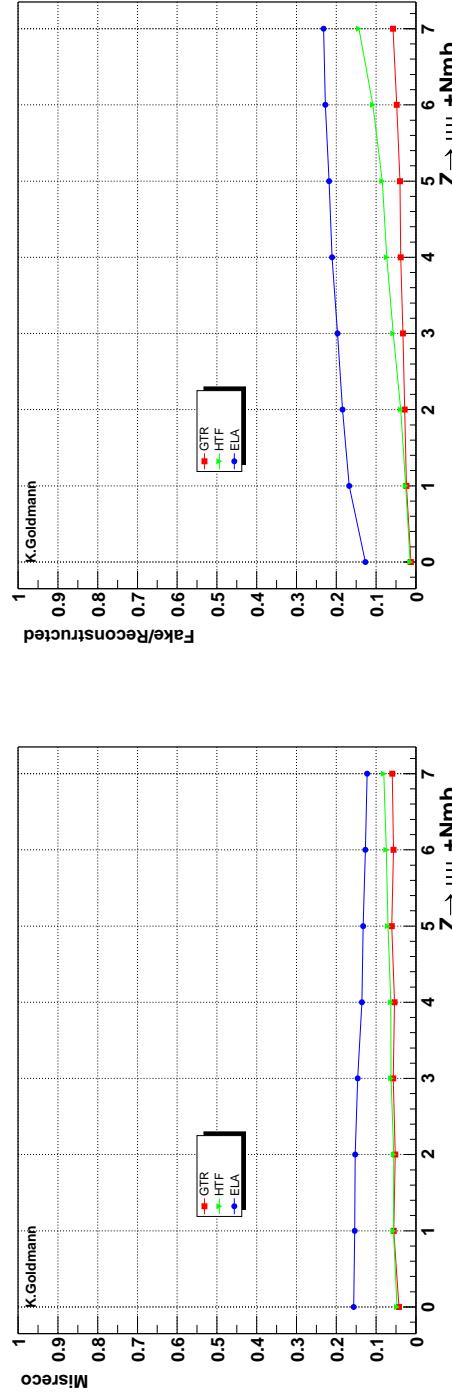
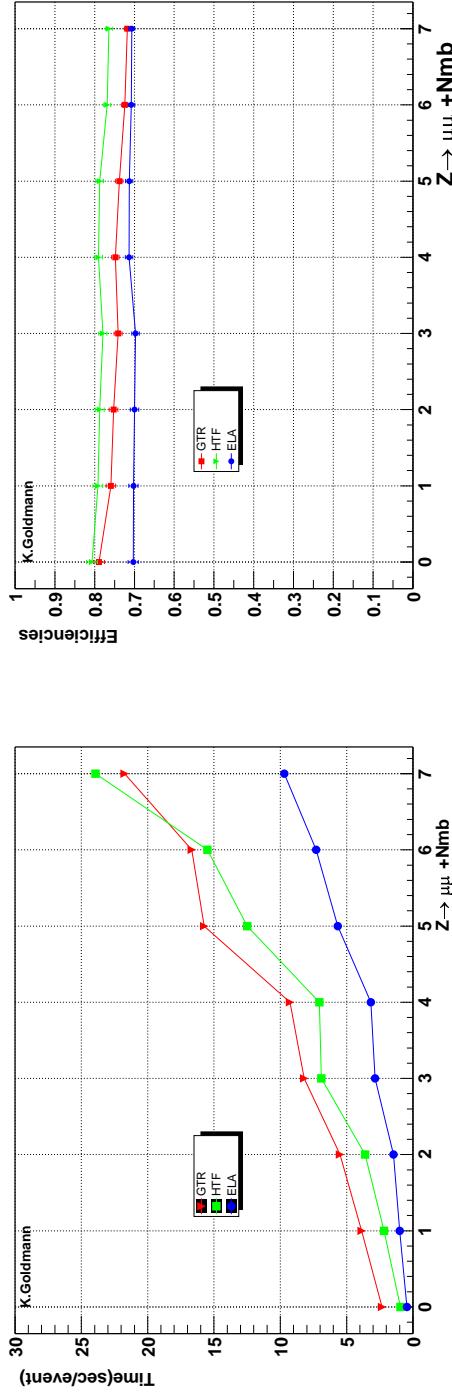
### Future plans:

- GTR performance in jets
- double track resolution studies
- feedback from physics groups (search for optimal algorithm)
- combination of different tracking (p12/p13)



## Algorithms Comparison

We have started to compare different track finders



Time, efficiency, fake rate, # of misreco as a function of Nmb.

Track reconstruction software is working

We develop/support four different tracking algorithms

Recent innovations in DØ global tracking include:

- significant improvements made in GTR timing performance
- design and implementation of global DØ multiple scattering propagator (unique refitting step for all tracking)
- development of track reconstruction in non uniform magnetic field, including track extrapolation to H disks
- track extrapolation between DØ subdetectors from DCA up to muon system
- track reconstruction in gap and overlap regions, most vulnerable parts of the DØ detector
- integration of four tracking algorithms into unique working structure
- hit mask for thumbnail
- global objects, ObjTable  $\leftrightarrow$  (cuts, propagators, mag. field update, etc.)
- design and implementation of analysis tools (`gtr_analyze` is part of `reco_analyze`)
- Matching based on hit-by-hit comparison

All studies used Monte Carlo events. Expect differences with real data.  
Areas of concern:

### Event Generation

Event generators may not match well with real events or even with each other. In our benchmark  $Z \rightarrow \mu\mu$  events ISAJET produces much more underlying event than PYTHIA. PYTHIA with one background event is roughly comparable to ISAJET with none.

### Detector Response

The response of the real detector may differ from our simulation (DOGSTAR plus digitization)

### Detector Inefficiencies

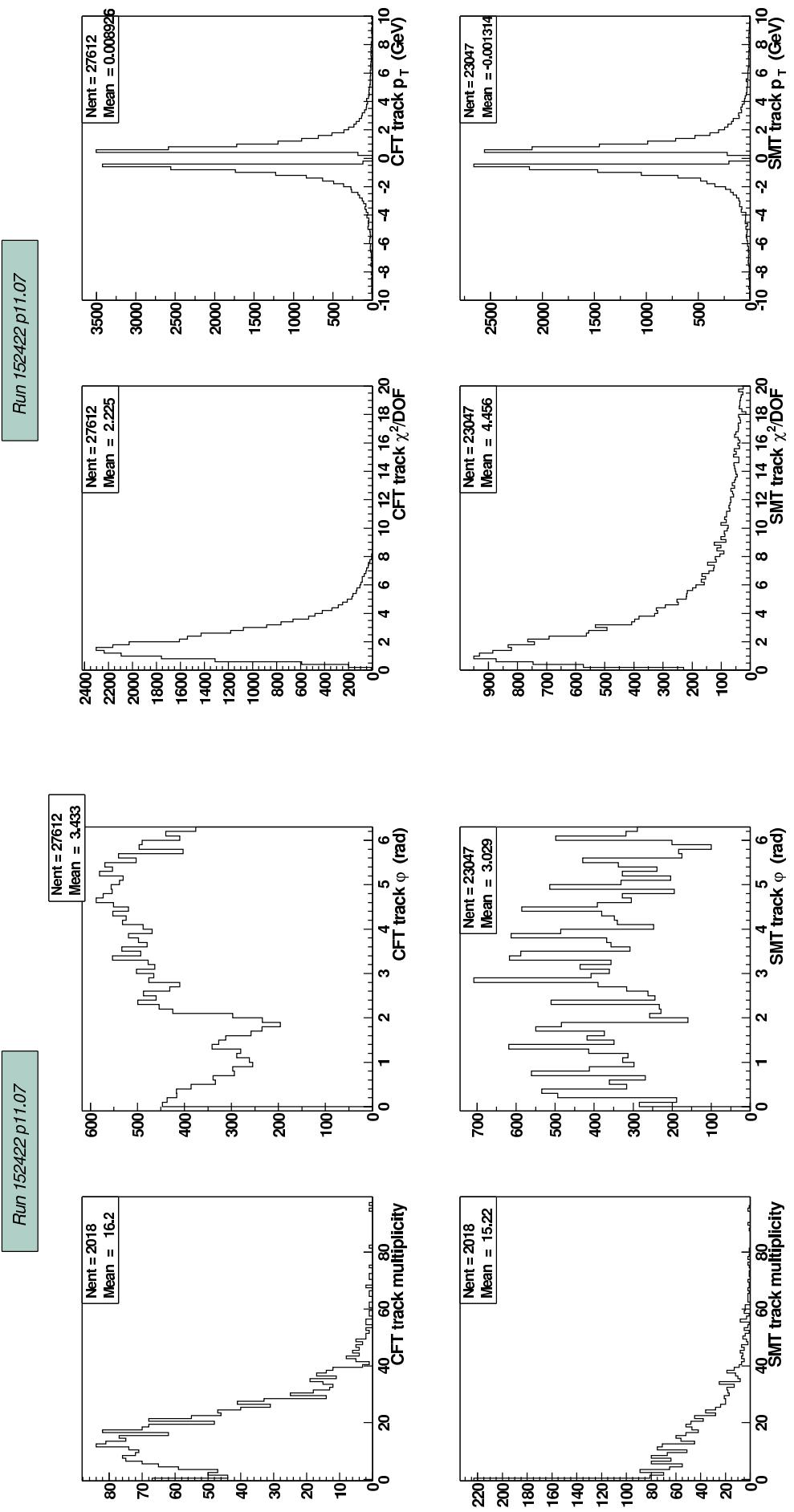
Reduced detector response, dead channels, lead to inefficiencies in detector elements. Present track-finding paths make optimistic assumptions about these efficiencies especially in the CFT. We can expect an increase in CPU time if we allow for these inefficiencies or some loss in track-finding efficiency if we do not.

### Detector Noise

If detectors have more noise than expected, then expect an increase in CPU time. If the noise is severe, resolution may be degraded.



## Global Tracking with Real Data (results) —



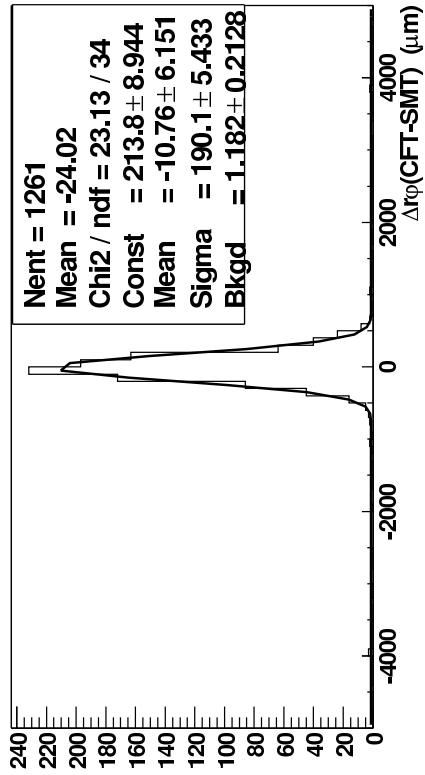
CFT/SMT track multiplicity and  $\phi$

CFT/SMT track  $\chi^2/\text{dof}$  and  $p_T$

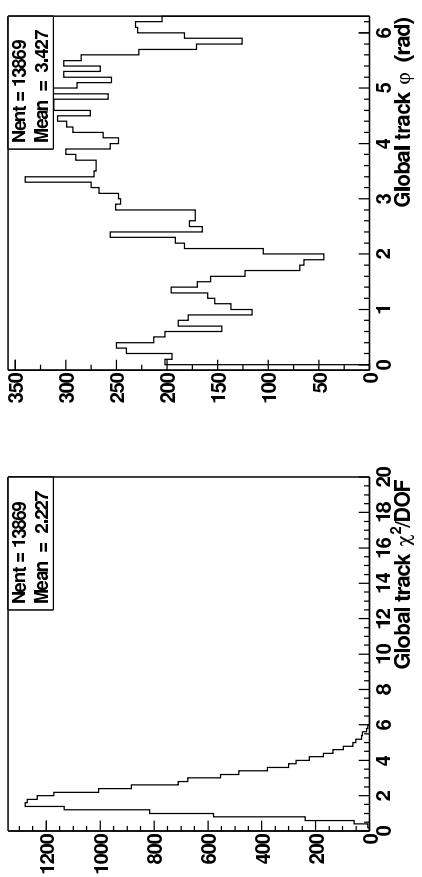
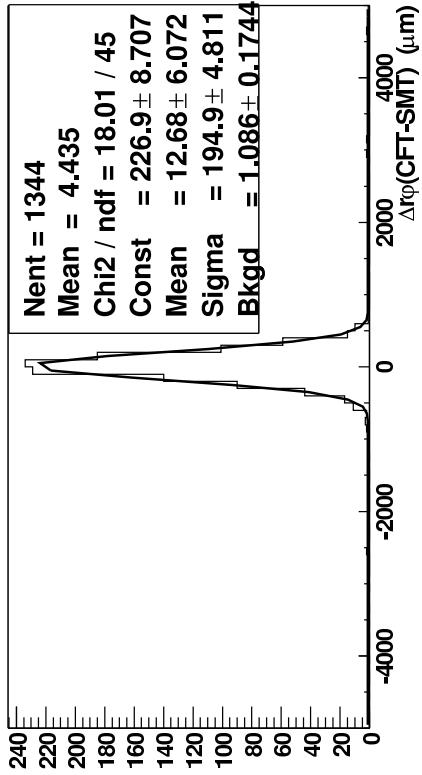
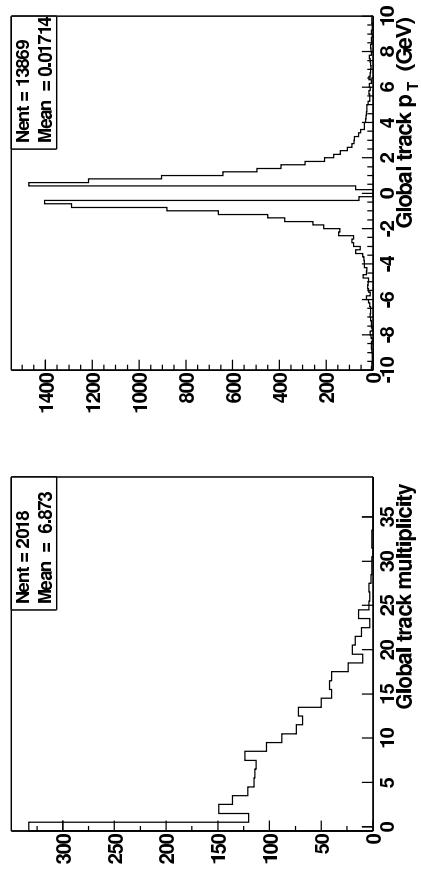


## Global Tracking with Real Data (results) —

Run 152422



Run 152422 p11.07



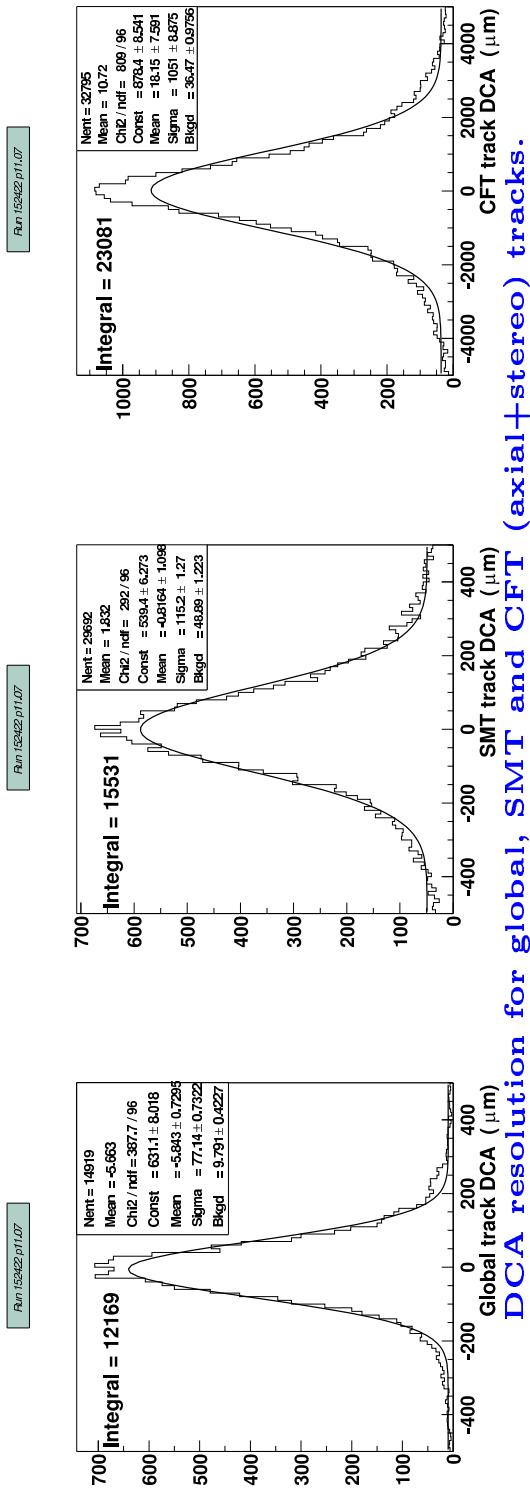
track matching at first CFT cylinder,

$$\sigma(\Delta r\phi) \sim 190 \mu\text{m}$$

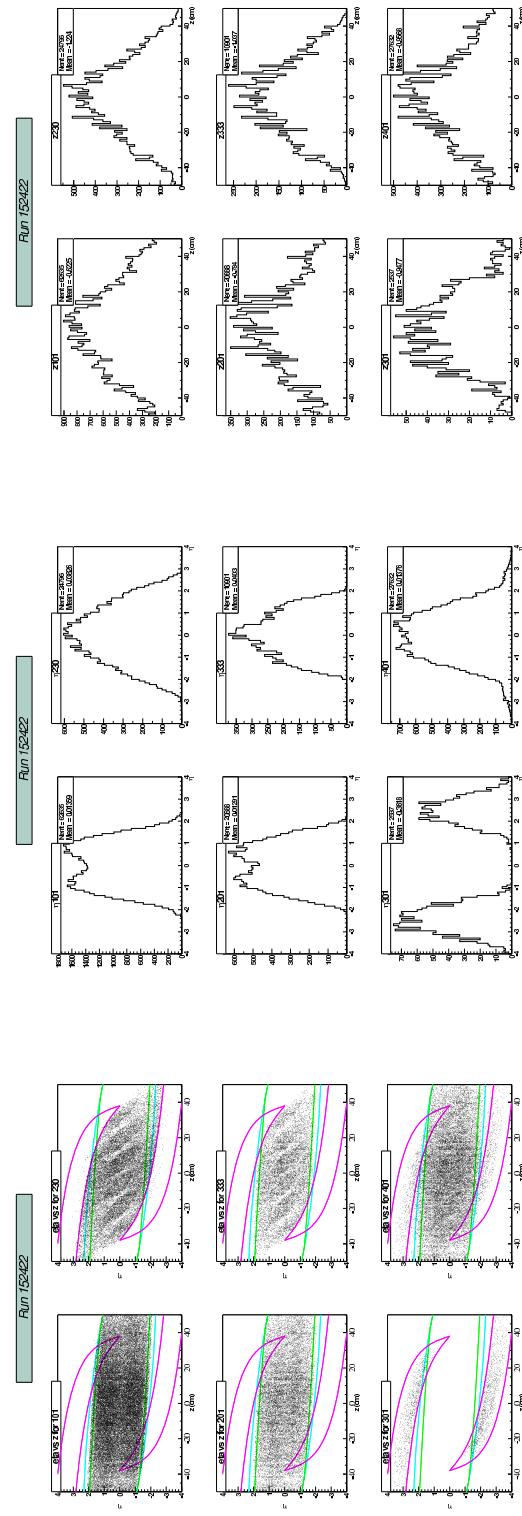
Global track multiplicity and  $\phi$



Global Tracking with Real Data (results)



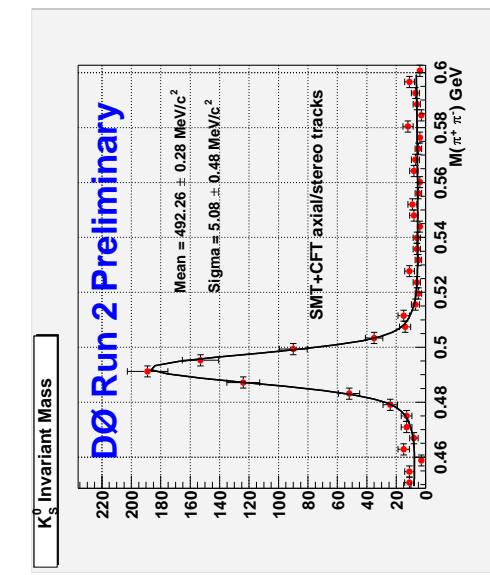
DCA resolution for global, SMT and CFT (**axial+stereo**) tracks.



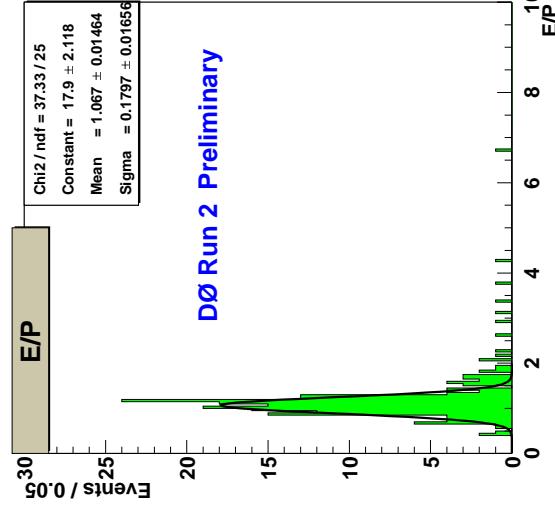
$\eta$  vs  $z$ ,  $\eta$  and  $z$  distributions for different chunks



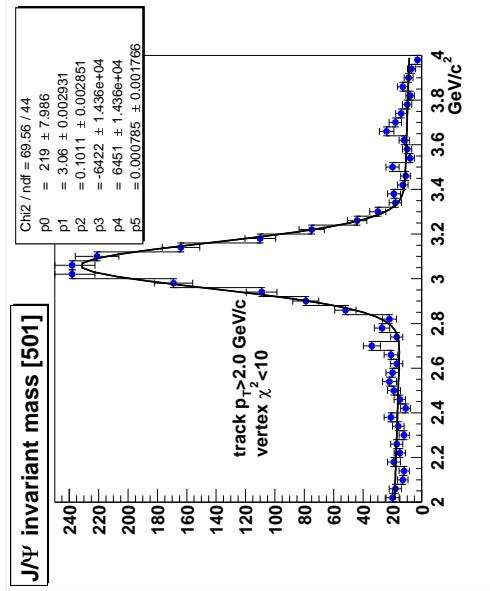
## Global Tracking with Real Data (physics) —



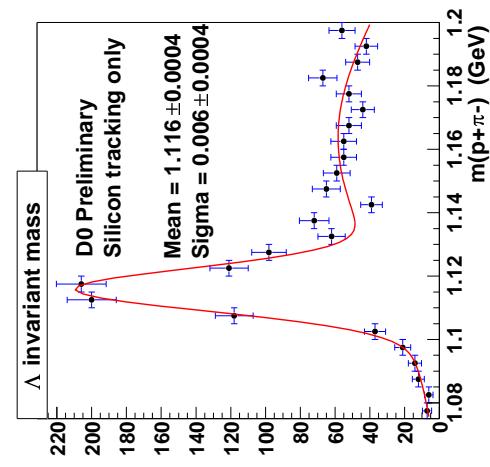
### global tracks



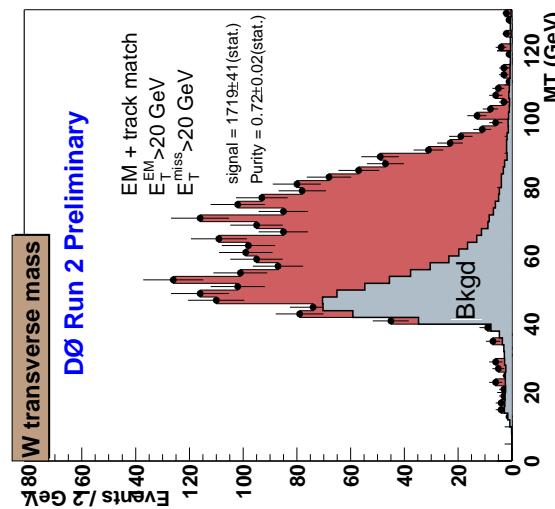
### track matching



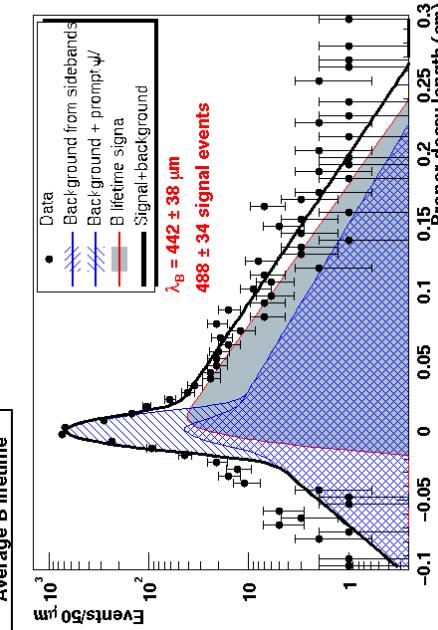
### gtr/htf combine



### AA tracking



### low pT physics



### high pT physics



- ◆ beam spot measurements:  $30 \pm 3\mu\text{m}$
- ◆ IP resolution (global tracks):  $62\mu\text{m}$  for  $p_T > 0.5 \text{ GeV}$ ,  $45\mu\text{m}$  for  $p_T > 3 \text{ GeV}$
- ◆ DCA resolution:  $77\mu\text{m}$  global,  $115\mu\text{m}$  SMT and  $1050\mu\text{m}$  CFT tracks
- ◆ tracks are found everywhere (central, overlap, gap, forward regions)
- ◆ average track multiplicity in data:  $\sim 16 \text{ trk/CFT}$ ,  $\sim 15 \text{ trk/SMT}$ ,  $\sim 6.8 \text{ trk/global}$
- ◆ flat  $\phi$  distribution
- ◆ CFT vs. SMT relative misalignment:  $\sigma[\Delta r \varphi(CFT - SMT)] \simeq 190\mu\text{m}$
- ◆  $\chi^2/dof \simeq 2.2$  for global tracks with flat  $\chi^2$  probability distribution
- ◆ Vertex resolution (with 4 tracks or more):  $35\mu\text{m}$
- ◆ Tracking has been used by many people and demonstrated that we do find physics objects:
  - ◆ DATA:  $\sigma(K_s) \sim 5 \text{ MeV}/c^2$
  - ◆ DATA:  $\sigma(J/\Psi) \sim 60 \text{ MeV}/c^2$
  - ◆ DATA:  $\sigma(\Lambda) \sim 6 \text{ MeV}/c^2$

We're approaching our MC performance predictions



## — Global Tracking with Real Data (p10.15.xx & p11.0x.xx) —

### p10.15.xx series

- first release used for data processing
  - SMT stand-alone reconstruction, CFT axial tracks
  - first appearance
    - SMT $\rightarrow$ CFT extension
    - CFT misses
    - GTR analyze package
  - modified tracking to handle arbitrary beam position
  - B field is fixed (-2T)
  - special data path to handle partial detector instrumentation
  - $\sim 6$  sec/event
- p11.0x.xx series
- many problems fixed
  - track multiplicity is fixed in both direction (SMT $\rightarrow$ CFT and CFT $\rightarrow$ SMT)
  - CFT wide doublets implementation
  - data path is pretty closed to MC
  - use data path over MC sample to estimate tracking efficiency
  - B field is dynamic, it is loaded from run number
  - first attempt to combine gtr/htf (for B-physics purposes, di-muon candidates)
    - $\sim 10$  sec/event (more complicated path, e.g. cover overlap region)

We made significant progress in tracking to allow physics groups to start serious analysis



## Global Tracking with Real Data (p12/p13 plans) —

- Choice of algorithm(s)
    - Compare performance of tracking algorithms (gtr, htf, elastic\_reco, AA)
    - Study how best to integrate non-gtr trackers into standard tracking to optimize performance & cpu time
  - Support for tilted beam axis
    - Add inefficiencies to simulation
  - Performance analysis. Define a new set of standard plots that can be produced from gtr\_analyze root tuples.
    - Physics Performance (as currently).
    - Findable track performance.
    - Tracking efficiency from data
  - Cluster algorithms
    - Incorporate ADC information into CFT clusters
    - SMT clusters
    - Update CFT cluster information from track z.
  - Tracking in roads (e.g. lepton candidates).
  - Monitor detector performance, alignment, calibration, commissioning, etc.
- 
- Improvements to gtr tracking*
- Merge MC and data paths.
    - Add CFT misses properly to chunks 113 & 111.
    - Replace chunks 211 & 333 with single cft extension chunk.
    - Add H disks to data paths.
    - Use new CFT clusters in 2D clusters.
  - Reduce inefficiencies.
- 
- Improvements in non-gtr trackers*
- (AA, htf, elastic\_reco)
    - Fake rate
    - Add misses
    - Phase space fiducial cuts
    - GTrack conversion

Track reconstruction software is a subset of DØ framework

It consists of four different tracking algorithms

Significant progress has been made to reconstruct our data by tracking and TTK groups

We start reconstruct our data as we reconstruct our MC

Recent innovations in DØ global tracking include:

- beam offset
- dynamic analysis tools
- run based mag. field and pedestal substruction
- CFT tracking with misses
- hit-by-hit comparison of reco'd and MC
- hitmask available in reco\_analyze
- a complete corrected CFT map (thanks to TTK group)
- speed up tracking code (by factor of 2 for real data)
- integration of gtr/htf/elastic tracking algorithms

Suggested release: p11.09.xx or development tree

Many thanks go to users for usefull feedback and criticism